[Continued on next page]

(54) Title: PRE-VERIFICATION OF CHECKSUMS USED WITH CHECKSUM-BASED HEADER COMPRESSION
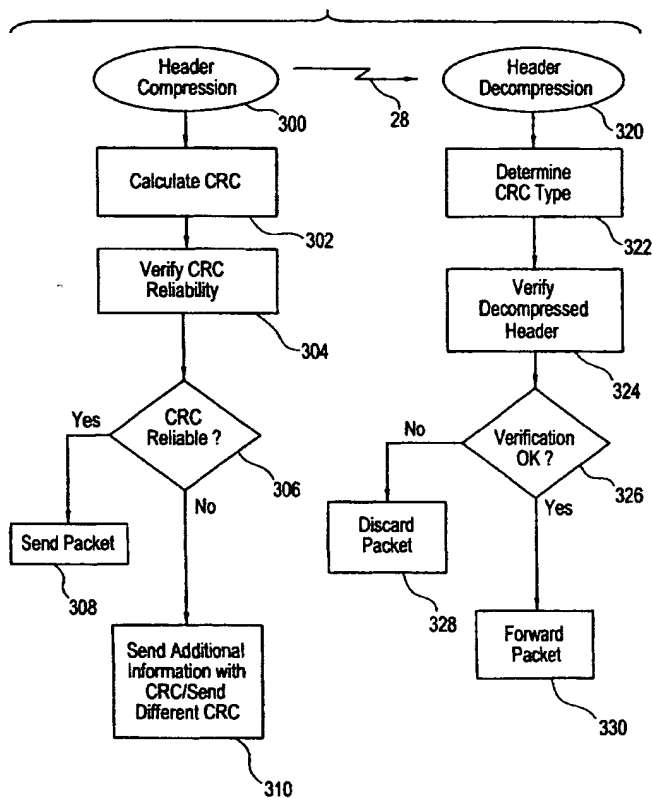
(57) Abstract: A method or apparatus for preverifying checksums ensures the checksums are reliable for header decompression (300) purposes. The checksums are pre-verified (304) on the compressor (50) side before being transmitted (308) to the decompressor (60). Verification (304) of the checksums includes testing against a number of packet loss scenarios to ensure reliable detection of errors. In addition, pre-verification (304) helps increase and/or maintain compression efficiency by keeping the length of checksums from becoming unnecessarily long.

patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

– *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

Published:

— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

-1-

PRE-VERIFICATION OF CHECKSUMS USED WITH
CHECKSUM-BASED HEADER COMPRESSION

CROSS-REFERENCES TO RELATED APPLICATIONS

This Application for Patent claims the benefit of priority from, and hereby incorporates by reference the entire disclosure of, co-pending U.S. Provisional Application for Patent Serial No. 60/188296 filed March 10, 2000.

BACKGROUND OF THE PRESENT INVENTION

Field of the Invention

The invention relates generally to packet communications and, more particularly, to header compression in packet communications.

Description of the Related Art

The tremendous success of the Internet has made it desirable to expand the Internet Protocol (IP) to a wide variety of applications including voice and speech communication. The objective is, of course, to use the Internet as a link for transporting voice and speech data. Speech data has been transported across the Internet using IP-based protocols such as the User Datagram Protocol (UDP) and the Real-time Transport Protocol (RTP) . In a typical application, a computer running telephony software converts speech into digital data which is then assembled into IP-based data packets that are suitable for transfer across the Internet. Additional information regarding the UDP and RTP protocols may be found in the following publications which are incorporated herein by reference: Jon Postel, User Datagram Protocol, DARPA RFC 786, August 1980; Henning Schulzrinne et al., RRT: A Transport Protocol for Realtime Applications, IETF RFC 1889, IETF Audio/video Transport Working Group, January 1996.

A typical IP-based speech data packet 10 is shown in FIG. 1. The packet 10 is one of a plurality of related packets that form a stream of packets 10 representing speech data being transferred over a packet-switched communication network such as the Internet. The packet 10 is made of a header portion 12 and a payload portion 14. The header 12 may include static information 16 such as the source and destination

-2-

addresses of the packet 10, and dynamic information 18 such as the IP identification, RTP sequence number, and RTP time stamp.

FIG. 2 illustrates a pertinent portion of an exemplary packet-switched communication network 20. A packet source 22 such as the Internet provides a stream of IP-based data packets 10 across a link 24 to an access technology 26 such as, for example, a base station. The access technology 26 compresses the data packets 10 received from the packet source 20 and transmits the compressed packets across a link 28 to a receiver 30 such as, for example, a mobile cellular station. The link 28 may be any radio interface between the access technology 26 and the receiver 30 such as, for example, a cellular link. The receiver 30 receives the compressed packets from the access technology 26 and decompresses the packets, including reconstruction of the packet headers.

In ordinary speech data transported over IP-based protocols, however, the header 12 may represent up to 70% of the data packet 10, leaving little capacity for the payload 14. For cellular links, this inefficient use of bandwidth is much too expensive for IP-based transportation to become a viable alternative to circuit switched speech services. Therefore, some compression or reduction of the header 12 is generally required.

The term header compression refers to the art of minimizing the necessary bandwidth for information carried in packet headers on a per hop basis over point-to-point links. The headers are compressed or otherwise reduced at the transmitting side and then reconstructed at the receiving side. As mentioned previously, headers generally comprise both static information and dynamic information, that is, information that may change from one packet to the next. Header compression is usually realized by sending only the static information initially. Dynamic information is then transferred by sending only the change, or delta, from the previous packet. Completely random information is usually sent without compression.

Under such a compression scheme, it is important to keep the states, also known as "context," of the various header fields consistent between the compressor and decompressor because reconstruction of the current packet depends on the context of the previous packets. For example, consider a packet that was somehow lost or

-3-

damaged during transmission from the compressor to the decompressor. The decompressor will not have been updated with the context of that packet. Consequently, any changes, or deltas, in the header fields of the next packet will be offset by one packet, and the decompressor may not be able to correctly reconstruct the header for the new packet.

There are generally three factors to consider in maintaining context consistency: (1) compression efficiency, (2) robustness, and (3) reliability. Compression efficiency relates to how much the headers are compressed and can generally be expressed by the average compressed header size. Robustness, on the other hand, relates to how well a compression scheme handles loss over a predefined link. For example, will a loss make the contexts inconsistent, possibly resulting in a large number of erroneously decompressed packet headers? Reliability relates to how trustworthy a scheme is in terms of catching incorrectly decompressed packets. Any header compression scheme, in order to be feasible, must take at least these factors into account.

One presently available header compression protocol is the ROCCO protocol (RObust Checksum-based header COmpression). ROCCO is a header compression protocol that uses checksums to verify the correctness of the reconstructed headers at the decompressor side. The particular checksum used is CRC (cyclical redundancy checking). In ROCCO, a CRC is calculated over the full original IP/UDP/RTP header by the compressor and then included in the compressed header. The compressed header (with the included CRC) is then sent to the decompressor where it is reconstructed. Once the header is reconstructed, a new CRC is calculated over the full reconstructed header. This new CRC is then compared with the included CRC to see if there is a match, which means the header reconstruction was performed correctly.

CRC calculation is well-known to those of ordinary skill in the art and will not be discussed here. Suffice it to say, however, the size of the CRC impacts the compression efficiency and reliability of the header compression scheme. For ROCCO, a CRC length of 10 bits is commonly used. In general, a long CRC provides higher reliability, but lower compression efficiency. A short CRC, on the other hand, provides higher compression efficiency, but lower reliability. As such, longer CRCs

-4-

are generally favored in order to ensure strong reliability, often at the expense of compression efficiency. It is desirable, therefore, to be able to generate a CRC that provides high reliability without compromising compression efficiency.

The present invention advantageously provides a CRC that has high reliability, yet does not unnecessarily compromise compression efficiency. In accordance with the invention, a CRC is pre-verified at the compressor before it is sent to the decompressor. The pre-verification ensures that the CRC is of sufficient length to provide robustness and reliability, but is not unnecessarily lengthy.

SUMMARY OF THE INVENTION

The present invention is directed to a method or apparatus for pre-verifying checksums to ensure the checksums are reliable for header decompression purposes. The checksums are preverified at the compressor side before being transmitted to the decompressor. Verification of the checksums includes testing against a number of packet loss scenarios to ensure reliable detection of errors. In addition, pre-verification helps increase and/or maintain compression efficiency by keeping the length of the checksums from becoming unnecessarily long.

In one aspect, the invention is related to a method of verifying a checksum in a header compression protocol. The method comprises the steps of calculating a checksum. based on a packet header, compressing the packet header in accordance with the header compression protocol, and testing the checksum. for reliability prior to including the checksum in the compressed packet header.

In another aspect, the invention is related to a system for verifying a checksum in a header compression protocol. The system comprises a checksum calculator for calculating a checksum based on a packet header, a compressor for compressing the packet header in accordance with the header compression protocol, and a reliability processor for testing a reliability of the checksum prior to including the checksum in the compressed packet header.

A more complete appreciation of the present invention and the scope thereof can be obtained from the accompanying drawings which are briefly summarized below, the following detailed description of the presently-pref erred embodiments of

-5-

the invention, and the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the method and apparatus of the present invention may be obtained by reference to the following Detailed Description when taken in conjunction with the accompanying Drawings wherein:

FIGURE 1 illustrates a typical speech data packet.;

FIGURE 2 illustrates a packet-switched communication environment;

FIGURE 3 illustrates a general packet flow according to an embodiment of the present invention;

FIGURE 4 illustrates a general flow of the checksum preverification process according to the embodiment of FIGURE 3;

FIGURE 5 illustrates a compressor apparatus according to the embodiment of FIGURE 3; and

FIGURE 6 illustrates a decompressor apparatus according to the embodiment of FIGURE 3.

DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED EXEMPLARY EMBODIMENTS

The present invention will now be described more fully hereinafter with reference to the accompanying drawings, in which preferred embodiments of the invention are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art.

As mentioned previously, ROCCO calculates a CRC over the full original IP/UDP/RTP header at the compressor and then includes that CRC in the transmitted compressed header. Once the compressed header is reconstructed at the decompressor, a new CRC is calculated over the full reconstructed header. This new CRC is then compared to the included CRC to see if there is a match, which means the reconstructed header is correct. If the verification fails (i.e., the CRCs do not match),

-6-

ROCCO provides for local repair of the reconstructed header by additional reconstruction attempts based on one or more assumptions about what may have happened during transmission of the packet. (Hence the term "robust" in the name of the protocol.) Examples of events that may have occurred during the transmission include reordering of the packets, damaged packets, and lost packets, all of which may result in a packet loss from the decompressor's perspective.

The first additional reconstruction attempt assumes there was no packet loss.

Subsequent reconstruction attempts assume a single preceding packet was lost, then two preceding packets lost, and so on. For each reconstruction attempt, the decompressor alternately assumes a zero, one, or two packet loss, and so on, and calculates a new CRC based on the most probable context assuming the packet (s) were lost.

The present invention takes advantage of the repair feature in ROCCO by pre-verifying the CRC before including and sending it with the compressed header. Figure 3 illustrates the general packet flow according to one embodiment of the present invention. At step 300, header compression is performed in accordance with the ROCCO protocol. A CRC is calculated over the full, original header at step 302. At step 304, the CRC is verified for reliability, a process which will be explained in more detail herein. If the CRC is reliable, step 306, the CRC is included with the packet and sent to the next destination, step 308. If the CRC is not reliable, than a different, reliable CRC is sent instead, or additional information such as extra header fields is sent with the packet at step 310.

Once the compressed header including the CRC is sent to the receiver via the link 28, it is reconstructed at step 320. At step 322, the type of CRC is determined, e.g., the standard CRC, or a modified version for CRCs that were not considered to be reliable.

At step 324, the reconstructed header is verified by calculating a new CRC for the reconstructed header and comparing it against the included CRC. If verification is confirmed, step 326, the decompressed packet including the reconstructed header is forwarded to the intended application. If verification is not confirmed, i.e., the newly calculated CRC does not match the included CRC, the packet is discarded at step 328.

-7-

As mentioned previously, reliability relates to the trustworthiness of a compression scheme. To this end, the CRC should be of a sufficient length to be able to catch most or all erroneous header reconstructions. The present invention considers two factors in calculating such a CRC: the order and assumptions made by the decompressor in its repair attempts, and the number of actual lost packets that can be tolerated, as specified at the receiver side.

Consider the exemplary case where only three lost packets can be tolerated, *i.e.*, the decompressor will assume up to three preceding packets were lost in its repair attempts before discarding the packet as damaged. Based on this scenario, the compressor of the present invention would calculate a CRC that meets the following "MUST FAIL" criteria:

Errors	Reconstruction attempts MUST FAIL if
One packet lost	CRCs calculated for 0 packet loss
Two packets lost	CRCs calculated for 0 packet loss
	CRCs calculated for 1 packet loss
Three packets lost	CRCs calculated for 0 packet loss
	CRCs calculated for 1 packet loss
	CRCs calculated for 2 packet loss

The above criteria maybe better understood with reference to the flow diagram of FIG. 4 in further illustration of the reliability verification process (shown in step 304) . At step 400, the calculated CRC is checked to see whether it can catch a 1 packet error if it is assumed that no packet loss occurred. In other words, if a true 1 packet error occurred during transmission and the decompressor began repair under the (wrong) assumption that there was no packet loss, generating a new CRC accordingly, would the calculated CRC catch (*i.e.*, not match) the newly generated CRC? If no, the compressor must recalculate the CRC at step 412, possibly resulting in a longer CRC with each recalculation. The check is repeated for up to a 3 packet error with an assumption of 2 packets loss, steps 402-410. If the CRC passes the last check, step 410, the process stops at step 414 and no more checks are performed according to this exemplary embodiment.

Once the calculated CRC passes all the checks, it is considered pre-verified for

-8-

reliability purposes and the length of the CRC is fixed. Thus, the present invention ensures the length of CRC will not be any longer than it needs to be for reliability purposes, thereby helping to at least maintain a high level of compression efficiency. However, if after a certain predefined number of recalculations have transpired or a certain predefined CRC length has been reached, the CRC is then considered to be unreliable. In that case, the CRC may still be included in the packet (step 310 in FIG. 3) if additional information about the CRC is included.

FIG. 5 illustrates an embodiment of the present invention as implemented in the access technology 26. The access technology 26 includes a packet compressor 50 for compressing a data packet such as the data packet 10 (shown in FIG. 1), including the header therein. A CRC calculator 52 calculates a CRC based on the original, uncompressed header of the received packet. A reliability processor 54 then checks the calculated CRC for reliability in accordance with the flow diagram illustrated in FIG. 4. If the CRC is sufficiently reliable, it is included with the compressed header, and the compressed packet is then sent to the receiver 30 via the transmitter 56. If not, additional information may be sent as explained previously.

Referring to FIG. 6, the receiver 30 includes a packet decompressor 60 for decompressing the compressed packet and reconstructing the header therein. A CRC extractor 62 extracts the included CRC and determines whether it is a standard CRC or a modified one (such as in the case of an unreliable CRC). A verification processor 64 calculates a new CRC based on the reconstructed header and then compares it to the included CRC. If there is a match, the header reconstruction process is considered to be correct, and the packet is sent to a codec 66 for further processing. If no match, then an error has occurred, and the verification processor 64 attempts to repair the error in accordance with the ROCCO protocol.

The previous description is of a preferred embodiment for implementing the invention, and the scope of the invention should not necessarily be limited by this description. The scope of the present invention is instead defined by the following claims.

-9-

WHAT IS CLAIMED IS:

1.   A method for verifying a checksum in a header compression protocol, the method comprising the steps of:

calculating a checksum based on a packet header;

compressing said packet header in accordance with said header compression protocol; and

testing said checksum for reliability prior to including said checksum in said compressed packet header.

2. The method according to claim 1, further comprising reconstructing said packet header and verifying said decompressed packet header using said checksum.

3. The method according to claim 2, further comprising reconstructing said packet header again in response to an erroneous previous reconstruction.

4. The method according to claim 1, further comprising modifying said checksum in response to said checksum failing said testing.

5. The method according to claim 1, further comprising including additional header information in said compressed header packet in response to said checksum failing said testing.

6.   The method according to claim 1, wherein said checksum is a CRC checksum.

7.   The method according to claim 1, wherein said header compression protocol is a ROCCO header compression protocol.

8. The method according to claim 1, wherein said testing comprises testing said checksum against a predefined number of actual packet errors, assuming a predefined number of packets lost.

-10-

9.   A system for verifying a checksum in a header compression protocol, the system comprising:

a checksum calculator for calculating a checksum based on a packet header;

a compressor for compressing said packet header in accordance with said header compression protocol; and

a reliability processor for testing a reliability of said checksum prior to including said checksum in said compressed packet header.

10.   The system according to claim 9, further comprising a decompressor for reconstructing said compressed packet header.

11.   The system according to claim 10, further comprising a verification processor for verifying said decompressed packet header using said checksum.

12. The system according to claim 10, wherein said verification processor repeats reconstruction of said packet header in response to an erroneous previous reconstruction.

13. The system according to claim 9, wherein said reliability processor modifies said checksum, in response to said checksum failing said testing.

14. The system according to claim 9, wherein said reliability processor includes additional header information in said header packer in response to said checksum. failing said testing.

15.   The system according to claim 9, wherein said checksum is a CRC checksum.
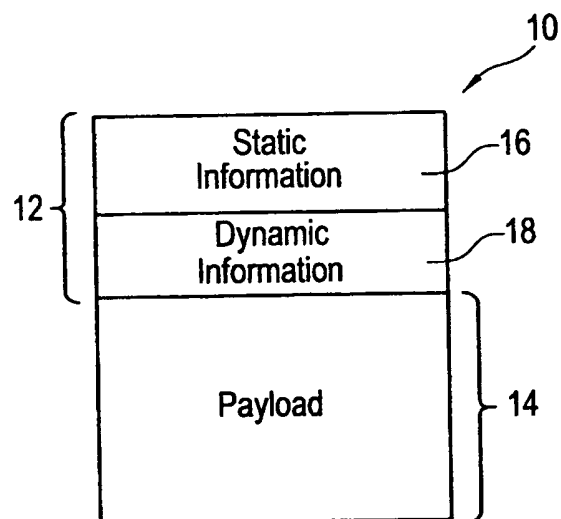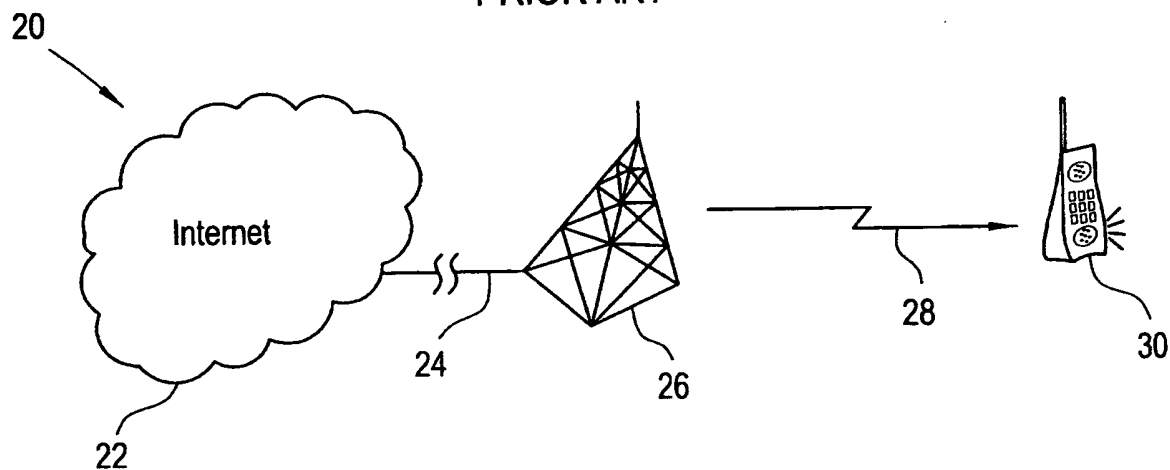
16.   The system according to claim 9, wherein said header compression

-11-

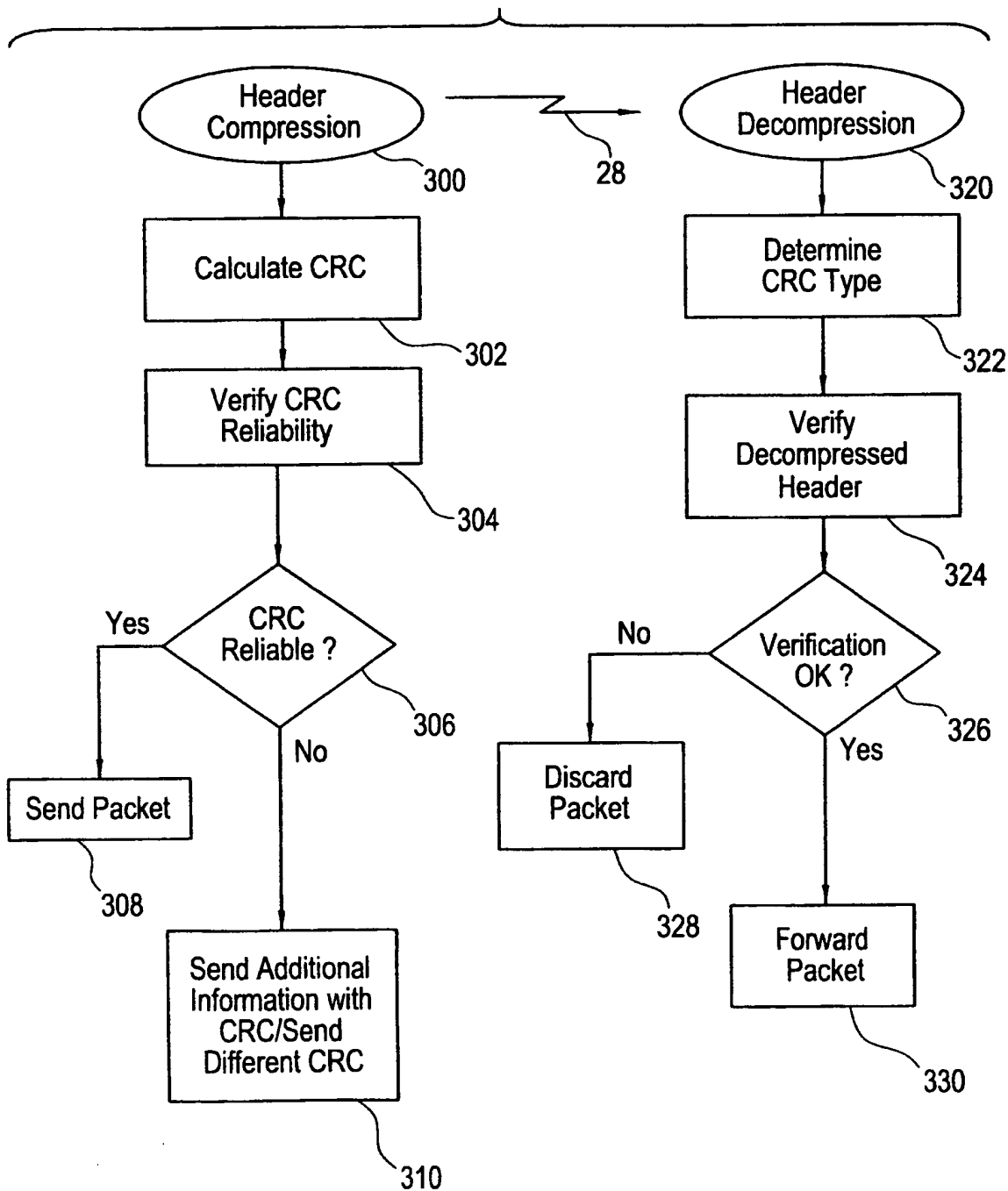protocol is a ROCCO header compression protocol

17. The system according to claim 9, wherein said reliability processor tests said checksum. against a predefined number of actual packet errors, assuming a predefined number of packets lost.
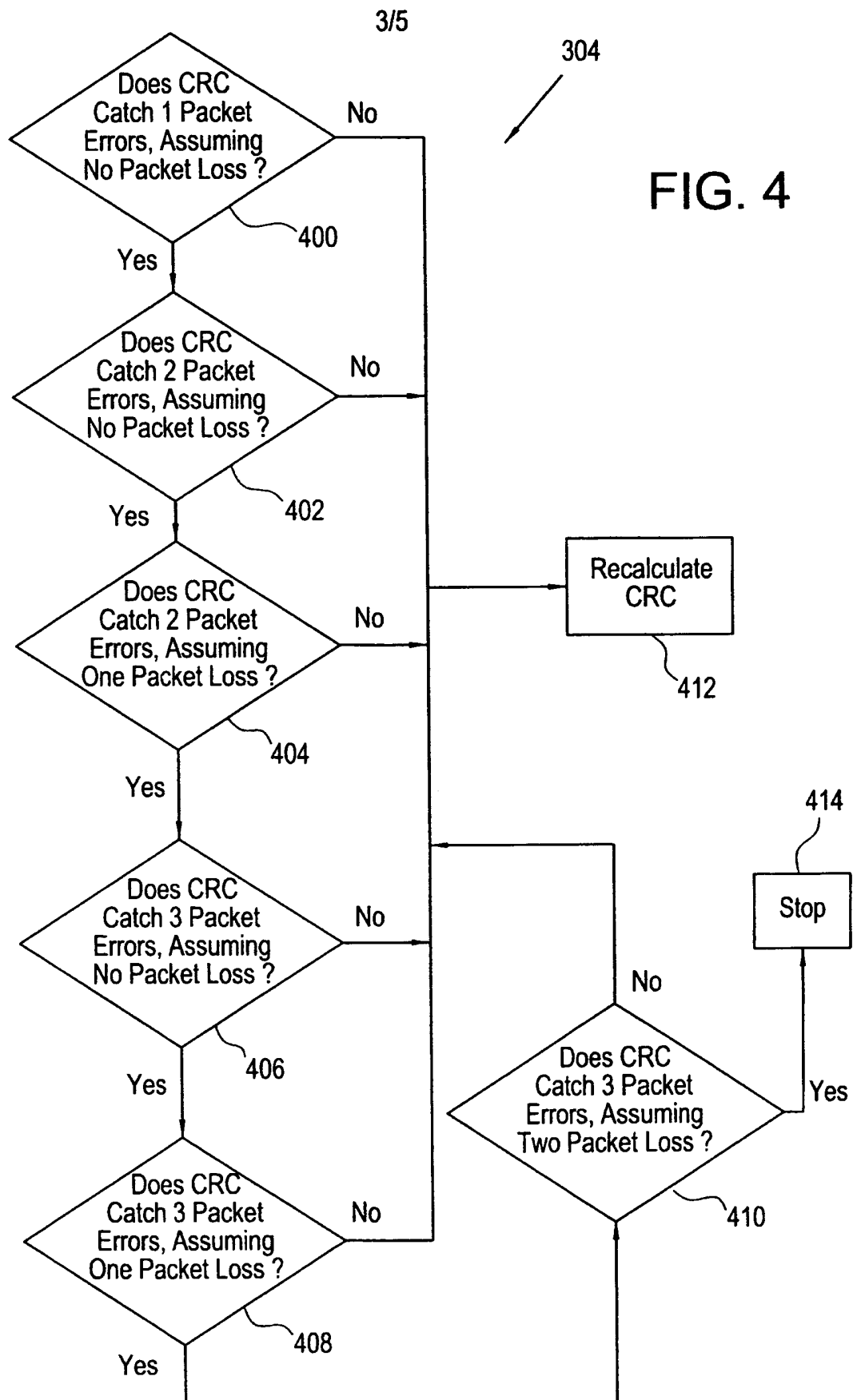
FIG. 1
PRIOR ART

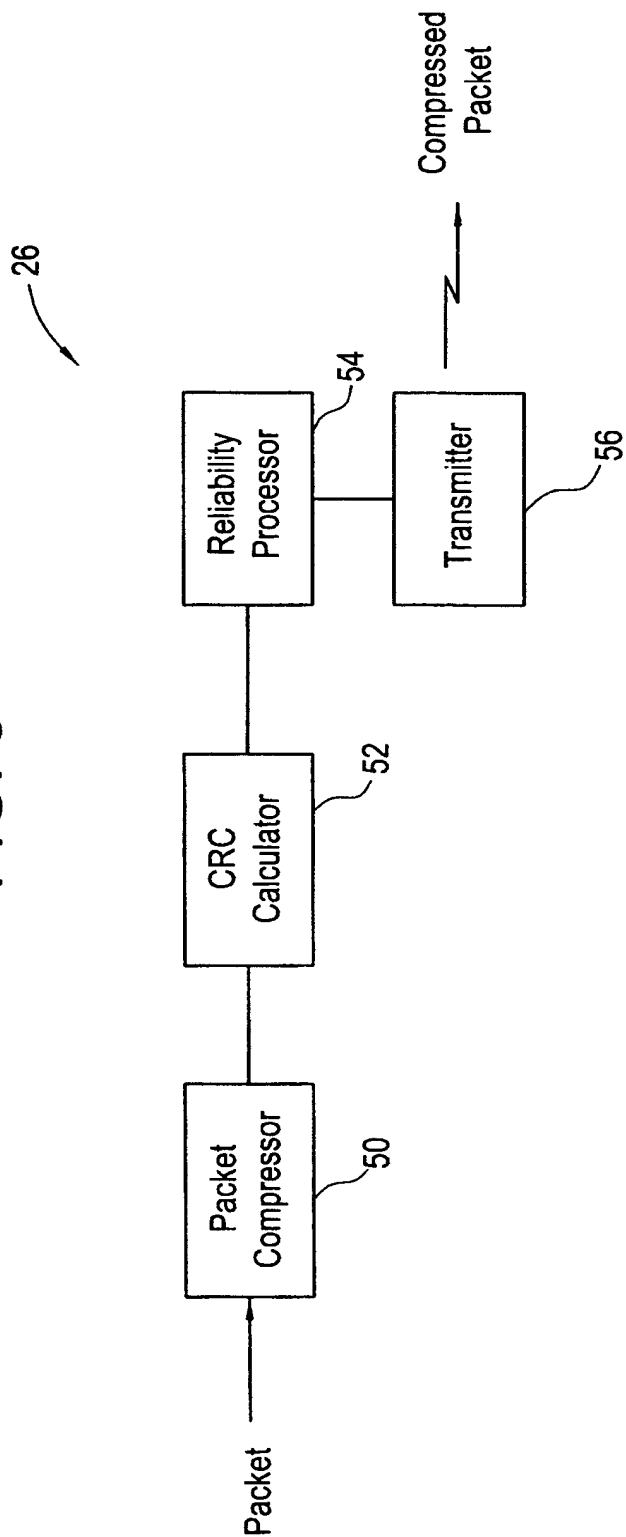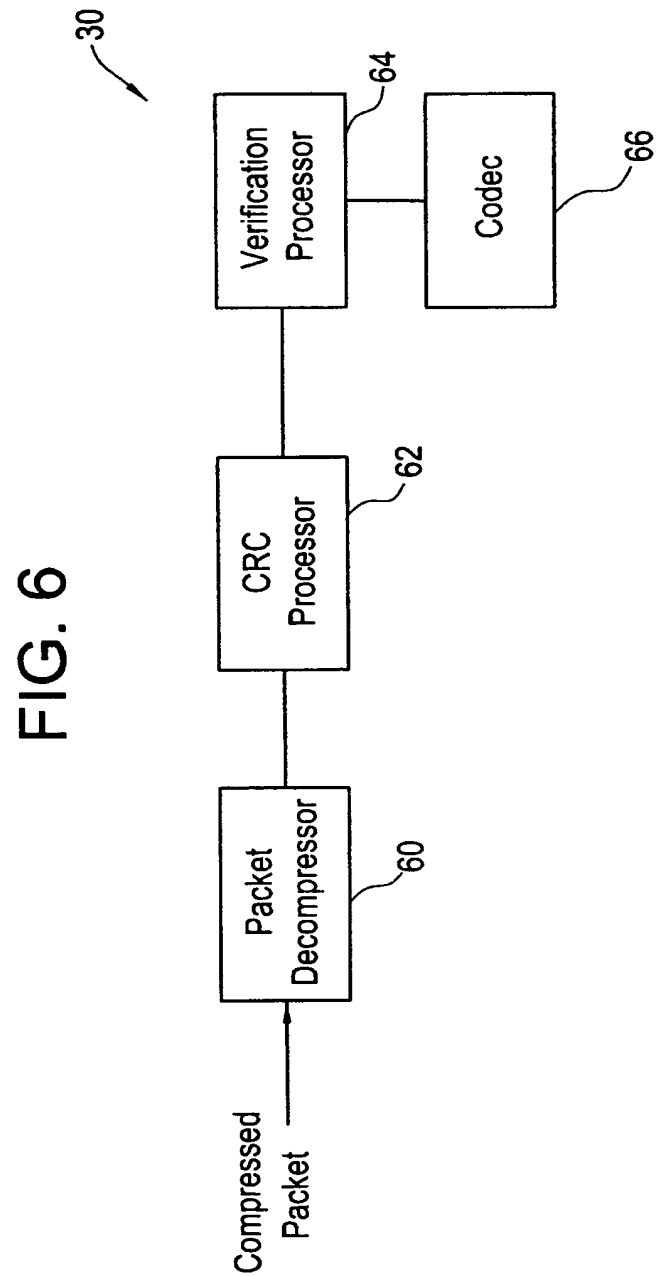| Static Information | 16 |
| Dynamic Information | 18 |
| Payload | 14 |

10

12

FIG. 2
PRIOR ART

20

22 Internet

24

26

28

30

2/5

FIG. 3

304

FIG. 4

Does CRC Catch 1 Packet Errors, Assuming No Packet Loss ? —No→

400

↓ Yes

Does CRC Catch 2 Packet Errors, Assuming No Packet Loss ? —No→

402

↓ Yes

Does CRC Catch 2 Packet Errors, Assuming One Packet Loss ? —No→

404

↓ Yes

Recalculate CRC

412

Does CRC Catch 3 Packet Errors, Assuming No Packet Loss ? —No→

406

↓ Yes

Does CRC Catch 3 Packet Errors, Assuming One Packet Loss ? —No→

408

↓ Yes

Does CRC Catch 3 Packet Errors, Assuming Two Packet Loss ?

410

No

Yes

Stop

414

FIG. 5

5/5

FIG. 6

Compressed
Packet →

Packet
Decompressor
60

CRC
Processor
62

Verification
Processor
64

Codec
66

30

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7   H04L29/06   H03M7/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7   H04L   H03M

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	LARS-ERIK JONSSON ET AL: "Robust checksum-based header compression (ROCCO)" NETWORK WORKING GROUP, INTERNET DRAFT , 10 March 2000 (2000-03-10), pages 1-67, XP002901828 the whole document	1-17
A	LARS-ERIK JONSSON ET AL : "Robust checksum-based header compression (ROCCO)" NETWORK WORKING GROUP, INTERNET-DRAFT, 22 June 1999 (1999-06-22), XP002901829 the whole document	1-17
A	V JACOBSON/1/ RFC1144: "Compressing TCP/IP headers for-low speed serial links" NETWORK WORKING GROUP, February 1990 (1990-02), XP002901830 the whole document	1-17

-/--

[X] Further documents are listed in the continuation of box C.	[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
31 July 2001	2 0. 08. 2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Peder Gjervaldsaeter

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 566 170 A (BAKKE MARK A ET AL) 15 October 1996 (1996-10-15) column 12, line 30 - line 49 column 9, line 55 -column 10, line 16 -----	1-17

INTERNATIONAL SEARCH REPORT

Intern   ial Application No

PCT/SE 01/00468

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5566170 A	15-10-1996	NONE	

Form PCT/ISA/210 (patent family annex) (July 1992)

This Page Blank (uspto)